REMARKS

Claims 1-12, 14-16, 18-29 and 31-33 are pending in the application. Claims 18-29 and 31-33 are amended. The amendments to the claims as indicated herein do not add any new matter to this application. Furthermore, amendments made to the claims as indicated herein have been made to exclusively improve readability and clarity of the claims and not for the purpose of overcoming alleged prior art.

CLAIM REJECTIONS—35 U.S.C. § 101

The Examiner's Answer mailed August 24, 2007 entered a new ground of rejection. The Examiner's Answer rejected Claims 18-29 and 31-33 under 35 U.S.C. § 101. The Applicants respectfully submit that the amendments to Claims 18-29 and 31-33 cure any deficiencies from which Claims 18-29 and 31-33 might have suffered under 35 U.S.C. § 101.

CLAIM REJECTIONS—35 U.S.C. §§ 102-103

A. The Features of Claims 1-4, 6, 18-21, and 23 Are Not Disclosed, Taught, or Suggested by Fox

Among other features, Claim 1 recites that a schema evolver receives a document that indicates changes **that are to be made to an existing ("first") XML schema**. The schema evolver **generates an evolved ("second") XML schema based on both (a) the existing XML schema and (b) the document that indicates the changes**.

Thus, Claim 1 is all about generating a new, evolved ("second") XML schema, and has nothing to do with transforming documents that conform to a schema into documents that conform to a different schema.

In contrast, Fox is all about transforming documents that conform to a schema into documents that conform to a different schema, and has nothing to do with generating a new, evolved XML schema.

Specifically, Fox describes generating an XSLT script **to transform documents that conform to one schema into documents that conform to a different schema**. Fox refers to the XSLT script as a “transformation.” Fox calls the mechanism that generates the XSLT script a “transformation generator.” Fox states that the transformation generator is “for **generating a transformation** from the first schema into the second schema” (paragraph [0072], last 4 lines).

Significantly, Fox doesn’t generate the second schema based on the first schema and the “transformation.” Instead, Fox generates the “transformation” based on the first schema (the “source data schema”) **and** the second schema (the “target data schema”). Because the transformation is generated based on the second schema, the “transformation” clearly cannot be generated until the second schema already exists. Since the transformation cannot be generated until the second schema already exists, it make no sense to say that the second schema is generated based on the transformation.

Fox describes how this “transformation” is generated as follows: “At step 120, a source data schema and a target data schema are imported” (paragraph [0104]). “At step 180, a transformation is derived for transforming data conforming with the source data schema into data conforming with the target data schema” (paragraph [0107]).

It is clear from this description that the “transformation,” which is “for transforming data that conforms to the source data schema into data that conforms to the target data schema” is “derived” from the source **and target** data schemas rather than the target data schema being derived from the source data schema and the “transformation.”

Additionally, Fox shows that schema receiver 210 and transformation generator 260 **receive, as input, both a source data schema and a target data schema**. Based on this input, transformation generator 260 **outputs a "derived transformation."**

Additionally, paragraph [0050] says that there is "a need for a tool that can transform data **conforming to** a first schema into data **that conforms to** a second schema." Paragraph [0049] discusses the problem of different companies using different **existing** schemas, and how this problem makes it difficult for these companies to use each other's data (because the data conform to different **existing** schemas).

Additionally, paragraph [0051] says that the "present invention provides a method and system **for deriving transformations** for transforming data from one schema to another." It is the **transformation** that is derived based on the **schemas** rather than a **schema** being derived based on the **transformation**. Paragraph [0051] also mentions that an XSLT script may be generated, and explains that the XSLT script can be applied to (a) documents that **conform to** the source XML schema in order to generate (b) documents that **conform to** the target XML schema. Clearly, the XSLT script transforms the documents that **conform to** the XML schemas rather than the XML schemas **themselves**.

Additionally, paragraph [0061] says, "**Given a source XML schema and a target XML schema** . . . an appropriate **transformation** of source to target XML documents **is generated**." Clearly, the **transformation** is generated based on the **source and target XML schemas** rather than the **target XML schema** being generated based on the **transformation**.

Thus, Fox is virtually brimming with statements that support the Applicants' position. Fox **does not** disclose, teach, or suggest that a schema evolver generates an evolved XML schema based on both (a) an existing XML schema and (b) a document that indicates changes that are to be made that are to be made to the existing XML schema, as required by Claim 1.

By virtue of their dependence from Claim 1, Claims 2-4, 6, 18-21, and 23 include the features of Claim 1 distinguished from Fox above. As a result, Claims 1-4, 6, 18-21, and 23 are patentable over Fox under 35 U.S.C. § 102(e).

B. The Features of Claims 12, 15, 16, 29, 32 and 33 Are Not Disclosed, Taught, or Suggested by Fox

Among other features, Claim 12 recites "wherein said one or more database object types were generated based on a second XML schema that differs from said first XML schema."

The Office Action alleges that Fox discloses this limitation in paragraph [0453], lines 8-11, which say: "For example, if the given table column has data type VARCHAR2, then the choice of properties may only include properties with target type string, or compositions of properties whereby the final property in the composition has target type string." The Office Action alleges that the "target type string" is the second XML schema, and that "VARCHAR2" is a database object type (Final Office Action, footnote 4).

However, "target type string" is not an XML schema in any way. As used in Fox, a "string" is a data type that comprises a sequence of one or more characters, as is well known in the art. As those skilled in the art are well aware, although an XML schema may comprise a "string," the mere fact that an XML schema comprises a "string" does not make every "string" an XML schema. Not every string has the qualities that an XML schema has. Therefore, the two are not identical or interchangeable.

Additionally, Claim 12 recites that the "database object types" must have been "generated based on" the "second XML schema." Thus, if Fox's "target type string" is taken to be analogous to the "second XML schema" of Claim 12, and if Fox's "VARCHAR2" is taken to be analogous to the "database object types" of Claim 12, then Fox's "VARCHAR2" must have been "generated based on" Fox's "target type string." Fox does not indicate that this is the case. Fox does not indicate that "VARCHAR2" (alleged "database object types") was "generated based on" any XML schema whatsoever. Actually, "VARCHAR2" is a well-known data type that comes built-in to a popular database system. Therefore, the Office Action's proposed analogy does not fit the method of Claim 12.

Fox does not disclose, teach, or suggest "wherein said one or more database object types were generated based on a second XML schema that differs from said first XML schema" as recited in Claim 12.

By virtue of their dependence from Claim 12, Claims 14-16, 29, 32, and 33 include the features of Claim 12 distinguished from Fox above. As a result, Claims 12, 14-16, 29, 32, and 33 are patentable over Fox under 35 U.S.C. § 102(e).

C. <u>The Features of Claims 14 and 31 Are Not Disclosed, Taught, or Suggested by Fox</u>

Claim 14 depends from Claim 12 and further recites, "wherein said first XML schema was generated based on said second XML schema."

The Office Action alleges that Fox discloses this feature of Claim 14 in paragraph [0502], which reads, in its entirety:

> Preferably, components are implemented as objects that can send and receive Messages to and from other objects. Thus, for example, an indirect property P_2OP_1 and an indirect inheritance (C, D) are implemented as their own objects. Direct dependencies among the objects are indicated by in FIG. 27 by directed

> directed edges within the dependency graph. If a first object depends on a second object either through a direct (single edge) or indirect (multiple edge) dependency, then modification or deletion of the second object potentially impacts the first object. For example, referring to FIG. 27, a constraint depends directly on an indirect property, and a mapping depends indirectly on an indirect property.

Although the above paragraph refers to dependencies between **objects**, the above paragraph does not indicate that these objects are different **XML schemas** or that these objects were generated **based on different XML schemas**. There appears to be absolutely no relation or similarity between the **objects** described in this paragraph and the **XML schemas** recited in Claim 14. Appellants concede that is it well known that one object may depend upon another object. However, it does not follow from this well-known fact that an evolved XML schema must be generated based on an existing XML schema.

As is discussed above with reference to Claim 1, Fox does not disclose, teach, or suggest generating an evolved XML schema based on an existing XML schema. Instead, Fox is concerned with transforming (a) documents that **conform** to one schema into (b) documents that **conform** to another schema. Fox's approach assumes that the two schemas already exist, and does not propose any approaches for generating either schema.

Therefore, Fox does not disclose, teach, or suggest "wherein said first XML schema was generated based on said second XML schema" as recited in Claim 14.

By virtue of its dependence from Claim 14, Claim 31 includes the features of Claim 14 distinguished from Fox above. As a result, Claims 14 and 31 are patentable over Fox under 35 U.S.C. § 102(e).

D. The Features of Claims 9 and 26 Are Not Disclosed, Taught, or Suggested by Fox

Claim 9 depends from Claim 1 and further recites, "wherein said one or more changes are expressed as one or more instances of **one or more XML types specified by a third XML schema**." For example, in paragraphs [0055]-[0058] of the present application, three XML types, "append-node," "insert-node-before," and "delete-node," are described. These XML types are specified in an "xdiff schema" which is separate from both the existing schema (e.g., existing XML schema 106) and the evolved schema (e.g., evolved XML schema 112). The XML document (e.g., XML document 108), which describes the changes that are to be made to the existing schema in order to produce the evolved schema, describes those changes in terms of XML change elements. Each of these XML change elements is an instance of an XML type (e.g., "append-node," "insert-node-before," or "delete-node") that is specified in the "xdiff" ("third") XML schema.

The Office Action alleges that Fox discloses this feature of Claim 9 in paragraph [0200], which reads, in its entirety:

> Reference is now made to FIGS. 11A-11R, which are illustrations of a for transforming data from one XML schema to another using the Coherence software application, in accordance with a preferred embodiment of the present invention. Shown in FIG. 11A is a window with package view of an Airline Integration ontology model in its left lane. The left pane displays classes from a fundamental package. A class Date is shown highlighted; and its properties are

> its properties are shown in the right pane. Fundamental packages are used for standard data types. Shown in FIG. 11B is a window with a hierarchical view of the Airline Integration ontology model in its left pane. The left pane indicates that FrequentFlyer is a subclass of Passenger, Passenger is a subclass of Person, and Person is a subclass of Being. The right pane displays general information about the class FrequentFlyer.

The above paragraph does **not** disclose, teach, or suggest that the changes that are to be made to one XML schema (e.g., existing XML schema 106) are expressed as instances of XML types (e.g., "append-node," "insert-node-before," and "delete-node") that are specified by another XML schema (e.g., the "xdiff schema").

Therefore, Fox does not disclose, teach, or suggest "wherein said one or more changes are expressed as one or more instances of **one or more XML types specified by a third XML schema**" as recited in Claim 9.

By virtue of its dependence from Claim 9, Claim 26 includes the features of Claim 9 distinguished from Fox above. As a result, Claims 9 and 26 are patentable over Fox under 35 U.S.C. § 102(e).

E. <u>The Features of Claims 5, 7-8, 22, and 24-25 Are Not Disclosed, Taught, or Suggested by Fox or Chau</u>

By virtue of their dependence from Claim 1, Claims 5, 7-8, 22, and 24-25 inherit the features of Claim 1 that are distinguished from Fox above. Therefore, Fox, taken individually, does not disclose, teach, or suggest the subject matter of any of Claims 5, 7-8, 22, and 24-25.

Chau also does not disclose, teach, or suggest the distinguished features of Claim 1 that are inherited by Claims 5, 7-8, 22, and 24-25. Indeed, the Office Action does not even allege that Chau discloses, teaches, or suggests these inherited features.

Consequently, even if Fox and Chau could be combined, the combination would still fail to to disclose, teach, or suggest that a schema evolver generates an evolved XML schema based on both (a) an existing XML schema and (b) a document that indicates changes that are to be made to the existing XML schema, as required by each of Claims 5, 7-8, 22, and 24-25.

As a result, Claims 5, 7-8, 22, and 24-25 are patentable over Fox and Chau under 35 U.S.C. § 103(a).

F. The Features of Claims 10, 11, 27, and 28 Are Not Disclosed, Taught, or Suggested by Fox or Vaitzblitz

By virtue of their dependence from Claim 1, Claims 10, 11, 27, and 28 inherit the features of Claim 1 that are distinguished from Fox above. Therefore, Fox, taken individually, does not disclose, teach, or suggest the subject matter of any of Claims 10, 11, 27, and 28.

Vaitzblitz also does not disclose, teach, or suggest the distinguished features of Claim 1 that are inherited by Claims 10, 11, 27, and 28. Indeed, the Office Action does not even allege that Vaitzblitz discloses, teaches, or suggests these inherited features.

Consequently, even if Fox and Vaitzblitz could be combined, the combination would still fail to disclose, teach, or suggest that a schema evolver generates an evolved XML schema based on both (a) an existing XML schema and (b) a document that indicates changes that are to be made to the existing XML schema, as required by each of Claims 10, 11, 27, and 28.

As a result, Claims 10, 11, 27, and 28 are patentable over Fox and Vaitzblitz under 35 U.S.C. § 103(a).

CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Please charge any shortages or credit any overages to Deposit Account No. 50-1302.

Respectfully submitted,

Hickman Palermo Truong & Becker LLP

Dated: February 1, 2008

/ChristianANicholes#50,266/
Christian A. Nicholes
Reg. No. 50,266

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone No.: (408) 414-1080
Facsimile No.: (408) 414-1076